CLMPTO

09/917638

07/31/01

DH

1. A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:
   - generating at least one of anti-latency data stream containing at least a leading portion of data for receipt by the client; and
   - generating at least one interactive data stream containing at least a remaining portion of said data for the client to merge into after receiving at least a portion of an anti-latency data stream.

2. The method of Claim 1, wherein:
   - said data is fragmented into K segments each requiring a time T to transmit over the network;
   - the anti-latency data streams includes M anti-latency data streams; and
   - the interactive data streams includes N interactive data streams.

3. The method of Claim 1, wherein:
   - the anti-latency data streams contains the leading portion of said data only;
   - the interactive data streams contains a whole set of said data.

4. The method of Claim 2, wherein:
   - each of the M anti-latency data stream contains substantially identical data repeated continuously within said anti-latency data stream, and wherein each successive anti-latency data stream is staggered by an anti-latency time interval; and
   - each of the N interactive data stream repeated continuously within said interactive data stream, and wherein each successive interactive data stream is staggered by an interactive time interval.

5. The method of Claim 4, wherein:
   - each of the M anti-latency data stream has J segments; and
   - the anti-latency time interval $\geq T$.

6. The method of Claim 4, wherein the interactive time interval $\geq JT$.

7. The method of Claim 5, wherein $M \geq J$.

8. The method of Claim 7, wherein $M = J$.

9. The method of Claim 6, wherein $N \geq \frac{R}{JT}$.

10. The method of Claim 9, wherein $N = \frac{R}{JT}$.

11. The method of Claim 8, wherein $M = N = J = \sqrt{\frac{R}{T}}$

12. The method of Claim 4, wherein each of the N interactive data streams contains the whole set of said data having K segments.

13. The method of Claim 4, wherein each of the N interactive data streams contains the remaining portion of said data only.

14. The method of Claim 4, further including the steps of:
- connecting the client to any one of the M anti-latency data streams when the client raises a request for said data; and
- connecting the client to any one of the N interactive data streams.

15. The method of Claim 2, wherein
- the anti-latency data streams includes:
  I. a leading data stream containing at least one leading segment of the leading portion of said data being repeated continuously within the leading data stream; and
  II. a plurality of finishing data streams, each of the finishing data streams:

- containing the rest of the leading portion of said data; and
- being repeated continuously within said finishing data stream, and wherein each successive finishing data stream is staggered by an anti-latency time interval;

- each of the N interactive data streams is repeated continuously within said interactive data stream, and wherein each successive interactive data stream is staggered by an interactive time interval.

16. The method of Claim 15, wherein
- each of the finishing data stream has J segments; and
- the anti-latency time interval $\geq T$.

17. The method of Claim 15, wherein the interactive time interval $\geq JT$.

18. The method of Claim 16, wherein $M \geq \frac{J}{2}+1$.

19. The method of Claim 18, wherein $M = \frac{J}{2}+1$.

20. The method of Claim 17, wherein $N \geq \frac{R}{JT}$.

21. The method of Claim 20, wherein $N = \frac{R}{JT}$.

22. The method of Claim 19, wherein $J = \sqrt{2K}$.

23. The method of Claim 15, wherein each of the N interactive data streams contains the whole set of said data having K segments.

24. The method of Claim 15, wherein each of the N interactive data streams contains the remaining portion of said data only.

25. The method of Claim 15, further including the steps of:

- connecting the client to the leading data stream when the client raises a request for said data;
- subsequently connecting the client to any one of the finishing data streams; and
- connecting the client to any one of the N interactive data streams.

26. The method of Claim 2, wherein:

- each of the N interactive data stream is repeated continuously within said interactive data stream, and wherein each successive interactive data stream is staggered by an interactive time interval $= \frac{KT}{N}$;
- the anti-latency data streams 1 to M are generated such that
  - an m^{th} anti-latency data stream has F_m segments, wherein F_m is an m^{th} Fibonacci number; and
  - the F_m segments are repeated continuously within the m^{th} anti-latency data stream.

27. The method of Claim 26, further including the steps of:

- connecting the client to at least the m^{th} and $(m+1)^{th}$ anti-latency data streams when the client raises a request for said data;
- buffering the data in at least the m^{th} and $(m+1)^{th}$ anti-latency data streams in the client;
- subsequently connecting the client to successive anti-latency data streams; and
- repeating the previous steps until all data in the leading portion is received by the client.

28. The method of Claim 27, further including the step of:

- connecting the client to any one of the N interactive data streams after all data in the leading portion is received by the client.

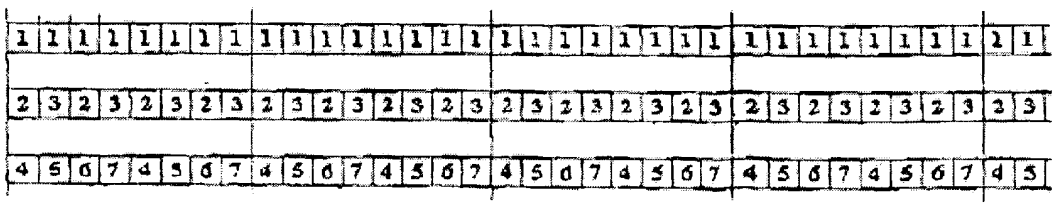
29. The method of Claim 26, wherein each of the N interactive data streams contains the whole set of said data having K segments.

30. The method of Claim 26, wherein each of the N interactive data streams contains the remaining portion of said data only.

31. The method of Claim 26, wherein $F_M \geq \frac{2K}{N}$.

32. The method of Claim 26, wherein m starts from 1.

33. The method of Claim 26, wherein m starts from 4 and the repeating 1st, 2nd, and 3rd anti-latency data streams have the following configuration:



34. The method of Claim 2, wherein:

- each of the N interactive data steams is repeated continuously within said interactive data stream, and wherein each successive interactive data stream is staggered by an interactive time interval $= \frac{KT}{N}$;
- in the M anti-latency data streams,
  I. the leading portion of said data contains 1 to J leading data segments labeled; and
  II. the leading data segments are distributed in the M anti-latency data streams such that an j^{th} leading segment is repeated by an anti-latency time interval $\leq jT$ within the anti-latency data streams.

35. The method of Claim 34, further including the steps of:

- connecting the client to all of the M anti-latency data streams when the client raises a request for said data; and
- buffering the leading portion of said data in the M anti-latency data streams in the client.

36. The method of Claim 35, further including the step of:

- connecting the client to any one of the N interactive data streams after all data in the leading portion is received by the client.

37. The method of Claim 34, wherein each of the N interactive data streams contains the whole set of said data having K segments.

38. The method of Claim 34, wherein each of the N interactive data streams contains the remaining portion of said data only.

39. The method of Claim 34, wherein $M \geq \sum_{j=1}^{j=J}(\frac{1}{j})$ and $J = \frac{K}{N}$.

40. The method of Claim 34 wherein six of the M anti-latency data streams containing the leading data segments are arranged as follows:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	32	2	4	2	8	2	4	2	16	2	4	2	8	2	4	2	64	2	4
3	6	9	3	12	42	3	6	18	3	24	9	3	6		3	12	48	3	6	9	3	36	39	3	6	18	3	12	9	3	6		3
5	10	15	25		5	20				5	10	30		55	5	40	15			5	10	45	50		5	20	55	25	60	5	10	15	
7	14	21	11	13	17	19	7	28	22	23	26	27	11	7	14	29	13	31	33	34	7	17	21	11	19	37	31	7	14	13	22	33	23
38	41		43	44		46	47	49				51	52				53		54			56	57		58		59		61		62	63	

wherein those segments in blank contains any data.

41. The method of Claim 2, wherein the M anti-latency data streams

- contains the leading portion of said data; and
- further includes two batches of data streams being a 1st set of anti-latency data streams and a 2nd set of anti-latency data streams.

42. The method of Claim 41, wherein:

- the 1^{st} anti-latency data streams have A 1^{st} anti-latency data streams from 1 to A, wherein
  I. an a^{th} anti-latency data stream has F_a segments, and F_a is an a^{th} Fibonacci number; and
  II. the F_a segments are repeated continuously within the a^{th} 1^{st} anti-latency data stream
- the 2^{nd} anti-latency data streams have B 2^{nd} anti-latency data streams, wherein each of the B 2^{nd} anti-latency data streams contains substantially identical data repeated continuously within said 2^{nd} anti-latency data stream, and wherein each successive 2^{nd} anti-latency data stream is staggered by a coarse-jump frame period;

such that the client can perform a coarse-jump function when the client is connected to the B 2^{nd} anti-latency data stream.

43. The method of Claim 42, further including the steps of:

- connecting the client to at least the a^{th} and $(a+1)^{th}$ 1^{st} anti-latency data streams when the client raises a request for said data;
- buffering the data in at least the a^{th} and $(a+1)^{th}$ 1^{st} anti-latency data streams in the client;
- subsequently connecting the client to successive 1^{st} anti-latency data streams;
- repeating the previous steps until all data in the A 1^{st} anti-latency data streams is received by the client.

43. The method of Claim 42, further including the steps of:

- connecting the client to at least the a^{th} and $(a+1)^{th}$ 1^{st} anti-latency data streams when the client raises a request for said data;
- buffering the data in at least the a^{th} and $(a+1)^{th}$ 1^{st} anti-latency data streams in the client;
- subsequently connecting the client to successive 1^{st} anti-latency data streams;
- repeating the previous steps until all data in the A 1^{st} anti-latency data streams is received by the client.

44. The method of Claim 43, further including the steps of:

- connecting the client to any one of the B 2^{nd} anti-latency data streams after all data in the 1^{st} anti-latency data streams is received by the client; and
- connecting the client to anyone of the N interactive data streams after all data in the connected B 2^{nd} anti-latency data stream is received by the client.

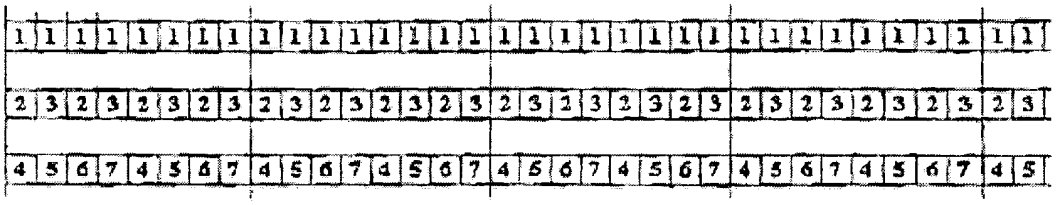
45. The method of Claim 42, wherein each of the N interactive data streams contains the whole set of said data having K segments.

46. The method of Claim 42, wherein each of the N interactive data streams contains the remaining portion of said data only.

47. The method of Claim 42, wherein said coarse-jump frame period includes E data segments, and $F_A \geq 2E$.

48. The method of Claim 42, wherein a starts from 1.

49. The method of Claim 42, wherein a starts from 4 and the repeating 1st, 2nd, and 3rd anti-latency data streams have the following configuration:



50. The method of Claim 41, wherein:

- the 1st anti-latency data streams have A 1st anti-latency data streams from 1 to A, wherein
  - I. an a^{th} anti-latency data stream has F_a segments, wherein F_a is an a^{th} Fibonacci number; and
  - II. the F_a segments are repeated continuously within the a^{th} 1st anti-latency data stream
- the 2nd anti-latency data streams have B 2nd anti-latency data stream including
  - I. a leading data stream containing at least one leading segment of the leading portion of said data being repeated continuously within the leading data stream; and
  - II. a plurality of finishing data streams, each of the finishing data streams:
    - containing the rest of the leading portion of said data; and

- being repeated continuously within said finishing data stream, and wherein each successive finishing data stream is staggered by a coarse-jump frame period

such that the client can perform a coarse-jump interactive function when the client is connected to the B 2^{nd} anti-latency data streams.

51. The method of Claim 50, further including the steps of:

- connecting the client to at least the a^{th} and $(a+1)^{th}$ 1^{st} anti-latency data streams when the client raises a request for said data;
- buffering the data in at least the a^{th} and $(a+1)^{th}$ 1^{st} anti-latency data streams in the client;
- subsequently connecting the client to successive 1^{st} anti-latency data streams;
- repeating the previous steps until all data in the A 1^{st} anti-latency data streams is received by the client.

52. The method of Claim 51, further including the steps of:

- connecting the client to the leading data stream after all data in the 1^{st} anti-latency data streams is received by the client;
- subsequently connecting the client to any one of the finishing data streams; and
- connecting the client to anyone of the N interactive data streams after all data in the B 2^{nd} anti-latency data streams is received by the client.

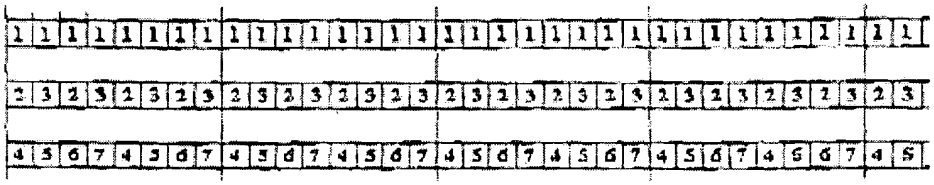
53. The method of Claim 50, wherein each of the N interactive data streams contains the whole set of said data having K segments.

54. The method of Claim 50, wherein each of the N interactive data streams contains the remaining portion of said data only.

55. The method of Claim 50, wherein said coarse-jump frame period includes E data segments, and $F_A \geq 2E$.

56. The method of Claim 50, wherein a starts from 1.

57. The method of Claim 50, wherein *a* starts from 4 and the repeating 1st, 2nd, and 3rd data streams of the *A* 1st anti-latency data streams have the following configuration:



58. The method of Claim 41, wherein:

- the 1st anti-latency data streams have *A* 1st anti-latency data streams, wherein,
  - I. the *A* 1st anti-latency data streams contains 1 to *C* 1st data segments; and
  - II. the 1st data segments are distributed in the *A* 1st anti-latency data streams such that an c^{th} leading segment is repeated by an anti-latency time interval $\leq cT$ within the *A* 1st anti-latency data streams;
- the 2nd anti-latency data streams have *B* 2nd anti-latency data streams, wherein each of the *B* 2nd anti-latency data streams contains substantially identical data repeated continuously within said 2nd anti-latency data stream, and wherein each successive 2nd anti-latency data stream is staggered by a coarse-jump frame period;

such that the client can perform a coarse-jump interactive function when the client is connected to the *B* 2nd anti-latency data stream.

59. The method of Claim 58, further including the steps of:

- connecting the client to all of the *A* 1st anti-latency data streams when the client raises a request for said data; and
- buffering data in the *A* 1st anti-latency data streams in the client until all data in the *A* 1st anti-latency data streams is received by the client.

60. The method of Claim 59, further including the steps of:

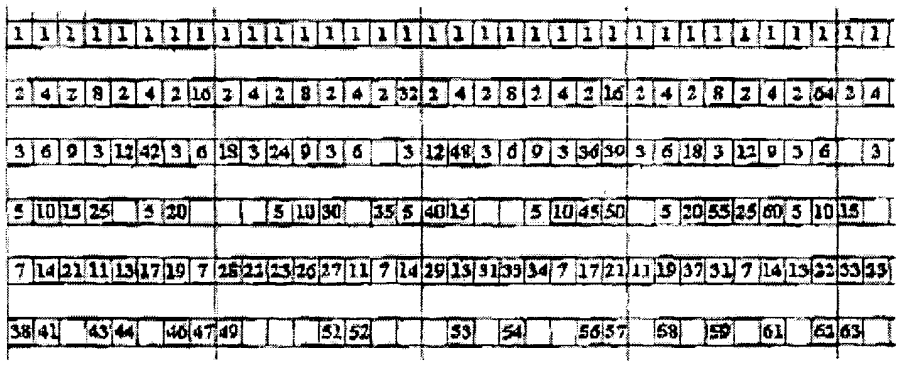
- connecting the client to any one of the B 2nd anti-latency data streams after all data in the 1st anti-latency data streams is received by the client; and
- connecting the client to anyone of the N interactive data streams after all data in the connected B 2nd anti-latency data stream is received by the client.

61. The method of Claim 58, wherein each of the N interactive data streams contains the whole set of said data having K segments.

62. The method of Claim 58, wherein each of the N interactive data streams contains the remaining portion of said data only.

63. The method of Claim 58, wherein said coarse-jump frame period includes E data segments, and $A \geq \sum_{c=1}^{c=E} (\frac{1}{c})$.

64. The method of Claim 58, wherein six of the A 1st anti-latency data streams are arranged as follows:



wherein those segments in blank contains any data.

65. The method of Claim 41, wherein:

- the 1st anti-latency data streams have A 1st anti-latency data streams, wherein,

I. the A 1st anti-latency data streams contains 1 to C 1st data segments; and

II. the data segments I are distributed in the A 1st anti-latency data streams such that an c^{th} leading segment is repeated by an anti-latency time interval $\leq cT$ within the A 1st anti-latency data streams;

- the 2nd anti-latency data streams have B 2nd anti-latency data stream including

I. a leading data stream containing at least one leading segment of the leading portion of said data being repeated continuously within the leading data stream; and

II. a plurality of finishing data streams, each of the finishing data streams:

- containing the rest of the leading portion of said data; and
- being repeated continuously within said finishing data stream, and wherein each successive finishing data stream is staggered by a coarse-jump frame period

such that the client can perform a coarse-jump interactive function when the client is connected to the B 2nd anti-latency data streams.

66. The method of Claim 65, further including the steps of:

- connecting the client to all of the A 1st anti-latency data streams when the client raises a request for said data; and
- buffering data in the A 1st anti-latency data streams in the client until all data in the A 1st anti-latency data streams is received by the client.

67. The method of Claim 66, further including the steps of:

- connecting the client to the leading data stream of the B 2nd anti latency data streams after all data in the 1st anti-latency data streams is received by the client;
- subsequently connecting the client to any one of the finishing data streams; and

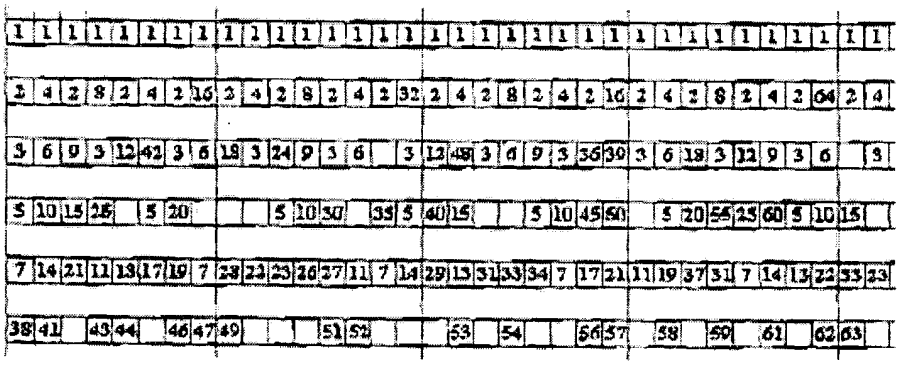
connecting the client to anyone of the N interactive data streams after all data in the B 2^{nd} anti-latency data stream connected in step F is received by the client.

68. The method of Claim 65, wherein each of the N interactive data streams contains the whole set of said data having K segments.

69. The method of Claim 65, wherein each of the N interactive data streams contains the remaining portion of said data only.

70. The method of Claim 65, wherein said coarse-jump frame period includes E data segments, and $A \geq \sum_{c=1}^{c=E} (\frac{1}{c})$.

71. The method of Claim 67, wherein six of the A 1^{st} anti-latency data streams are arranged as follows:



72. The method of Claim 2, wherein each of the K data segments contains a head portion and a tail portion, and the head portion contain a portion of data of the tail portion of the immediate preceding segment to facilitate merging of the K data segments when received by the client.

73. The method of Claim 2, further including the step of pre-fetching at least a portion of data in the leading portion in the client.

74. A method for transmitting data over a network to at least one client including the step of fragmenting said data into K data segments each requiring a time T to transmit over the network, wherein each of the K data segments contains a head portion and a tail portion, and the head portion contains a portion of data of the tail portion of the immediate preceding segment to facilitate merging of the K data segments when received by the client.

75. A method for transmitting data over a network to at least one client having a latency time to initiate transmission of said data to the client, including the steps of:

- generating at least one of anti-latency data stream containing at least a leading portion of data for receipt by the client;
- pre-fetching the leading portion in the client as pre-fetched data; and
- generating at least one interactive data stream containing at least a remaining portion of said data for the client to merge into the leading portion.

76. The method of Claim 75 further including the step of refreshing the pre-fetched data during a off-peak period.

77. The method of Claim 76, wherein the refresh time period is an off-peak period.

78. The method of Claim 76, wherein pre-fetched data is refreshed once per day.

79. A method for transmitting data over a network to at least one client including the steps of generating a plurality of anti-latency data streams, the anti-latency data streams include:

- a leading data stream containing at least one leading segment of a leading portion of said data being repeated continuously within the leading data stream; and
- a plurality of finishing data streams, each of the finishing data streams:
  - containing at least the rest of the leading portion of said data; and
  - repeated continuously within said finishing data stream, and wherein each successive finishing data stream is staggered by an anti-latency time interval.

80. The method of Claim 79 further including the steps of:
- connecting the client to the leading data stream when the client raises a request for said data; and
- subsequently connecting the client to any one of the finishing data streams.

81. The method of Claim 79, wherein said data is fragmented into K segments each requiring a time T to transmit over the network, and the anti-latency time interval $\geq T$.

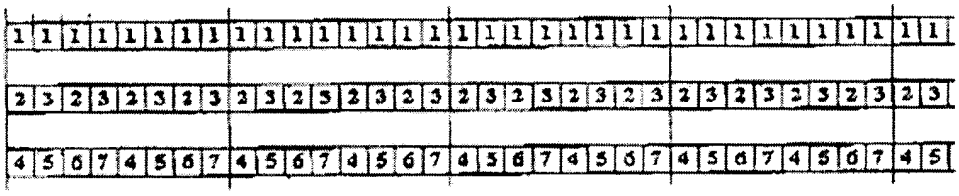
82. A method for transmitting data over a network to at least one client including the steps of:
- generating M anti-latency data streams from 1 to M, wherein an m^{th} anti-latency data stream has F_m segments, and F_m is an m^{th} Fibonacci number; and wherein said F_m segments are repeated continuously within the m^{th} anti-latency data stream.

83. The method of Claim 82 further including the steps of:
- connecting the client to at least the m^{th} and $(m+1)^{th}$ anti-latency data streams when the client raises a request for said data;
- buffering the data in at least the m^{th} and $(m+1)^{th}$ anti-latency data streams in the client;

- subsequently connecting the client to successive anti-latency data streams; and
- repeating the previous steps until all data in the leading portion is received by the client.

84. The method of Claim 82, wherein m starts from 1.

85. The method of Claim 82, wherein m starts from 4 and the repeating 1st, 2nd, and 3rd anti-latency data streams have the following configuration:



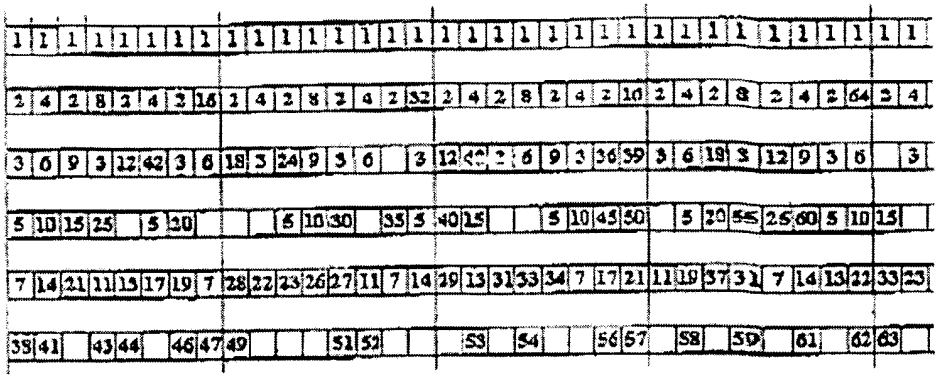
86. A method for transmitting data over a network to at least one client, said data being fragmented into K segments each requiring a time T to transmit over the network, including the steps of:

- generating M anti-latency data streams containing 1 to K anti-latency data segments, wherein the anti-latency data segments are distributed in the M anti-latency data streams such that an k^{th} leading segment is repeated by an anti-latency time interval $\leq kT$ within the anti-latency data streams.

87. The method of Claim 86 further including the steps of:

- connecting the client to all of the M anti-latency data streams; and
- buffering said data in the M anti-latency data streams in the client when the client raises a request for said data.

88. The method of Claim 86, wherein six of the M anti-latency data streams containing the leading data segments are arranged as follows:



wherein those segments in blank contains any data.

89. A method for receiving data being transmitted over a network to at least one client according to Claim 2, including the steps of:

- raising a request for said data; and
- connecting the client to the *M* anti-latency data streams and receiving data in the *M* anti-latency data streams.

90. The method of Claim 89 further including the steps of

- connecting the client to the *N* interactive data streams after all data in the *M* anti-latency data streams is received by the client.

91. The method of Claim 89, wherein data in the leading portion is received sequentially.

92. The method of Claim 89, wherein the client connects to at least two of the anti-latency data streams simultaneously.

93. The method of Claim 92 further including the steps of:

- buffering data in the two anti-latency data streams connected to the client that is received by the client sequentially.

94. The method of Claim 89, wherein the client connects to all of the anti-latency data streams simultaneously.

95. The method of Claim 94 further including the steps of:

- buffering data in the anti-latency data streams connected in the client; and
- rearranging the buffered data according to a proper sequence.

96. The method of Claim 89 further including the step of pre-fetching at least a portion of data in the M anti-latency data streams in the client as pre-fetched data.

97. The method of Claim 96 further including the step of refreshing the pre-fetched data during a refresh time period.

98. The method of Claim 97, wherein the refresh time period is 01:00-06:00.

99. The method of Claim 97, wherein the refresh time period is 10:00-15:00.

100. A method for receiving data being transmitted over a network to at least one client, wherein said data includes a leading portion and a remaining portion, and the remaining portion is transmitted by at least one interactive data stream including the steps of:

- pre-fetching the leading portion in the client as pre-fetched data; and
- merging the pre-fetched data to the remaining portion.

101. The method of Claim 100 further including the step of refreshing the pre-fetched data during a refresh time period.

102. The method of Claim 101, wherein the refresh time period is an off-peak period.

103. The method of Claim 101, wherein pre-fetched data is refreshed once per day.

104. The method of Claim 10, wherein $M = N = J = \sqrt{\frac{R}{T}}$

105. The method of Claim 21, wherein $J = \sqrt{2K}$.